

AMENDMENTS TO THE CLAIMS

1. (Original) A computer having an operating system including a kernel and a command interpreter installed therein, and operative to run an external command program, the kernel being operative to manage at least a file system and a memory and perform a multitasking control, the command interpreter being operative to input thereto a command character string indicative of one or more commands including external commands and interpret the command character string to execute the respective commands in the command character string in accordance with a control scheme specified by the character string, the external command program being described in an executable file stored in a predetermined directory of the file system, the program being executable by a process generated therefor by the kernel under control of the kernel, the command interpreter being implementable as one of the external commands,

wherein the external commands include a function command which is executable with a file name taken as an argument,

wherein, when the function command is executed with the file name taken as the argument, a process executing the function command inputs a script character string described in a predetermined format in a file specified by the file name, then converts the script character string into a command character string interpretable by the command interpreter, and newly invokes the command interpreter to output the command character string to the command interpreter.

2. (Original) A computer having an operating system including a kernel and a command interpreter installed therein, and operative to run an external command program, the kernel being operative to manage at least a file system and a memory and perform a multitasking control, the command interpreter being operative to input thereto a command character string indicative of one or more commands including external commands and interpret the command character string to execute the respective commands in the command character string in accordance with a control scheme specified by the character string, the external command program being described in an executable file stored in a predetermined directory of the file system, the program being executable by a process generated therefor by the kernel under control of the kernel, the command interpreter being implementable as one of the external commands,

wherein the external commands include a function command which is executable with a function index of a predetermined format taken as an argument,

wherein, when the function command is executed with the function index taken as the argument, a process executing the function command inputs a script character string describing an entity of the function index preliminarily stored in a memory accessible by the process, then converts the script character string into a command character string interpretable by the command interpreter, and newly invokes the command interpreter to output the command character string to the command interpreter.

3. (Original) A computer having an operating system including a kernel and a command

interpreter installed therein, and operative to run an external command program, the kernel being operative to manage at least a file system and a memory and perform a multitasking control, the command interpreter being operative to input thereto a command character string indicative of one or more commands including external commands and interpret the command character string to execute the respective commands in the command character string in accordance with a control scheme specified by the character string, the external command program being described in an executable file stored in a predetermined directory of the file system, the program being executable by a process generated therefor by the kernel under control of the kernel, the command interpreter being implementable as one of the external commands,

wherein the external commands include a function command which is executable with a file name or a function index of a predetermined format taken as an argument,

wherein, when the function command is executed with the file name taken as the argument, a process executing the function command inputs a main script character string described in a predetermined format in a file specified by the file name, then converts the main script character string into a command character string interpretable by the command interpreter, and newly invokes the command interpreter to output the command character string to the command interpreter,

wherein, when the function command is executed with the function index taken as the argument, a process executing the function command inputs a function script character string describing an entity of the function index preliminarily stored in a memory accessible by the process, then converts the function script character string into a command character string

interpretable by the command interpreter, and newly invokes the command interpreter to output the command character string to the command interpreter.

4. (Original) A computer as set forth in claim 3,

wherein the main script character string and the function script character string can be written in a file specified as the argument of the function command, and a command name of the function command can be described in these script character strings with a function name defined in the function script character string being taken as the argument,

wherein, when the function command is executed with the file name taken as the argument, the process executing the function command requests the kernel to allocate a shared memory for interprocess communication, and stores in the allocated shared memory the function script character string described in the file specified as the argument of the function command,

wherein, if the function name is described in the command character string before the command character string is outputted to the newly invoked command interpreter, the function name is replaced with a function index of the predetermined format specifying an address of the shared memory in which the function script character string associated with the function name is stored.

5. (Original) A computer as set forth in claim 3,

wherein, when the function command is executed with the function index taken as the argument, the process executing the function command requests the kernel to newly allocate a

shared memory for inter-process communication, copies the function script character string in the newly allocated shared memory from a shared memory allocated by the function command executed with the file name taken as the argument, converts the copied function script character string into a command character string, and outputs the converted command character string to the newly invoked command interpreter.

6. (Original) A computer as set forth in claim 3, wherein metacharacters for the command interpreter are usable in the main script character string.

7. (Original) A computer as set forth in claim 6, wherein one of the metacharacters causes the command to be executed in a background process.

8. (Original) A computer-readable storage medium containing thereon a program for a function command interpretable and executable by a command interpreter of an operating system, the function command being adapted to specify at least one argument at execution thereof, the program performing the steps of:

- (1) determining whether the argument is a file name or a function index;
- (2) if the argument is the file name, inputting a script character string including at least a main function portion and optionally a sub-function portion from a file specified by the file name, then converting the main function portion of the script character string into a command character string interpretable by the command interpreter and, if the sub-function portion is

present in the script character string, storing the sub-function portion in a shared memory for inter-process communication, and newly invoking the command interpreter to output the command character string to the command interpreter;

(3) if the argument is the function index, reading the sub-function portion in a shared memory specified by the function index, then converting the sub-function portion into a command character string interpretable by the command interpreter, and newly invoking the command interpreter to output the command character string to the command interpreter.

9. (Original) A computer-readable storage medium as set forth in claim 8, wherein an argument to be used by a program described in the form of a script character string can additionally be described in the file name or the function index serving as the argument of the function command.

10. (Original) A computer-readable storage medium as set forth in claim 8, wherein the function command has a return value transferable to the command interpreter.

11. (Original) A computer-readable storage medium as set forth in claim 8,
wherein a function name describable in the script character string can be defined in the subfunction portion,

wherein the program for the function command further performs the step of, if the function name is described in the command character string before the command character string

is outputted to the newly invoked command interpreter, replacing the function name with a function index of a predetermined format specifying an address of the shared memory in which a function script character string associated with the function name is stored.

12. (Original) A computer-readable storage medium as set forth in claim 8,
wherein the program for the function command, if the argument is the function index, reads the sub-function portion in the shared memory specified by the function index, newly allocates a shared memory to copy the read sub-function portion in the newly allocated shared memory, converts the sub-function portion into a command character string interpretable by the command interpreter, and newly invokes the command interpreter to output the converted command character string to the command interpreter.

13-14. (Cancelled)